CAROT 2/GENERIC 2

TEXT EDITOR PROGRAM

CENTRALIZED AUTOMATIC REPORTING ON TRUNKS (CAROT)

CONTENTS

NOTICE
Not for use or disclosure outside the
Bell System except under written agreement

CONTENTS PAGE

CONTENTS PAGE

1. INTRODUCTION

1.01 This section describes the text editor program available in CAROT 2, generic 2, software. This program is accessible via the CAROT 2 controller console or remote-user terminal locations. Before reading this section, the user should become familiar with the information contained in Section 190-102-203, Data Base Description and Input File Preparation, and Section 190-102-206, SELEC Program.

1.02 Whenever this section is reissued, the reason for reissue will be presented in this paragraph.

1.03 In the operation of the CAROT data base, it is sometimes necessary to perform large bulk changes to the information. These changes may not necessarily involve many records but do require a considerable amount of operator time to enter identification information which is unchanged in the operation. In order to ease this problem of data base management, the text editor was developed.

1.04 The main purpose of the text editor is to provide a means of editing (making changes to) information in the CAROT data base(s). The edited data is then used for input to the CAROT update processes which changes the data base. Since the text editor is intended for data base manipulation, the commands are slanted to that purpose. However, since it is a generalized editor, it may be used to perform edits on any ASCII (American Standard Code for Information Interchange) data.

1.05 The text editor is also used in conjunction with the CAROT report program generator (RPG/CAROT) to assist in the preparation of RPG/CAROT programs (see Section 190-102-207).

2. TEXT EDITOR COMMANDS

A. General

2.01 As previously mentioned, the CAROT text editor can be used to perform edits on any ASCII data but it is mainly intended to be used to alter data obtained via the SELEC program or for entry of RPG source programs.

2.02 The editor, used by itself or in conjunction with the SELEC program, should be used for all manual updates to the data base. Thus, providing update information via punched paper tape will no longer be required. For very small changes, the EDIT program may be used to simply enter a few lines of update information in a file. Larger changes may be more easily performed by first selecting a part of the data base and then editing in the changes.

Work-Space File

2.03 The data to be edited must be inputted to the text editor. The text editor can accept data from a peripheral device (such as magnetic tape, paper tape, and CRT or remote-user terminal) or from a disc file. When inputted, this data is loaded into an area on disc called the work-space file. This is the file in which all editing is performed. When data is loaded from a file, the original file is not disturbed unless the user specifically purges it or saves data into it.

> ***Note:*** As mentioned before, the EDIT work-space file is on disc. This work space file is given the name '$*nnn***A** and '$*nnn***B** where *nnn* is the remote-user number. The work-space file names will be listed along with the other EDIT files when using the **DL** (directory list) command.

2.04 Data stored in the work-space file is not saved when the user exits the text editor. Therefore, before exiting the text editor, the user should insure that his edited data is either saved on some peripheral device or stored in a disc file, by means of the **SAVE** command.

Character Strings

2.05 A character string is a sequence of ASCII characters. For example, ABCD is a character string. Many of the text editor commands allow the user to enter a character string qualified with column numbers. For example, "ABCD" (1Ø;23) indicates character string ABCD anywhere in columns 1Ø through 23. ABCD in any other column would be ignored. If a string is entered without qualifying column numbers, the string may occur anywhere in the line to be considered. The defaults are column number 1 and the length of the line. The following are valid entries:

ABCDE
HIJK(5;18)
WXYZ(5;62)

2.06 The text editor provides up to three labels which may be assigned by the user to a string of characters. These labels are $1 — $3 and are initially set to null and zero length. Once the user has set one of these labels to a character string (see **SET** command), the label may be used in place of the character string in the text editor commands. This can save the user much typing in operations which require repetitive inputs of a common string.

Don't Care Character

2.07 A don't care (DC) character can be used in a character string to skip over characters which are not of interest. The DC character is initially set to an at sign (@) but can be changed via the **SET:DC** command. The character skipped over will have no affect on the decision of whether or not the character string matches. On exchange operations, the DC character is used to indicate that the character in that position in the original string should be left and not replaced.

For example, the character string ABCD@@G where @ is the DC character would match the string ABCDEFG or the string ABCDJKG. If the string 1234@@7 was exchanged via the **XCHANGE** command with ABCDEFG, the result would be ABCDEFG and 1234EF7.

Line Designations

2.08 Each entry in the work-space file is assigned an integer line number from 1 through 9999 so that the first entry is assigned line number 1, and each succeeding line number is one more than the previous. Lines inserted between two lines are assigned fractional values indicating their position in the file. For example, two

lines inserted after line 35 would be referred to as line 35.1 and 35.2. When listing, only the fractional portion of the inserted line's line number will be outputted. An example is as follows:

```
35 ABCDEF
.1 ZYXW
.2 T1234
36 MNOP
```

where lines **ZYXW** and **T1234** were inserted after line number 35.

2.09 Line designation entries in commands may be specified in several ways: a single number, a number and the fractional part, an asterisk (*) to indicate the current line (last line displayed), a pound sign (#) to indicate the last line, a * or # with a numeral offset (*+6, # −8, etc), a character string, or columns 3 through 1Ø. All of the following are valid line designations:

1Ø1	*+3
1Ø1.12	*+5.1
*	#−6
#	"ABCD"
	"ABCD"(5;12)

Whenever a character string is used as a line designation, it should be enclosed in quotes to avoid confusion with numeric values. The quote character is indicated by the " symbol, but can be changed via the **SET:QUOTE=** command.

Current Line

2.10 Several commands which list, modify, or save lines of text can use the current line designation. Usually the form of these commands is to perform some function over a range. The range may be specified as line numbers or as character strings. Line numbers may be specified as absolute or relative. *Absolute* means the actual line, and *relative* means some line displaced by a constant from a specified position. For example, 35;4Ø would be an absolute notation indicating a range of lines from 35 through 4Ø. The range *;*+8 would indicate a range of lines from the current line to eight lines beyond the current line. The current line, identified by entry of an asterisk (*), is defined as the last line displayed. For example, if the operator listed lines 1 through 87 in the text file, then the current line would be 87 since it was displayed last. To determine which entry is the current line, simply output that line (see **WRITE** command).

B. Execution

2.11 The text editor is executed by typing **RUN:EDIT** followed with a carriage return. A 4-character prompter, **EDT?**, will appear indicating that the user is communicating with the text editor. To exit from the text editor, type **END** followed with a carriage return. Operation of the EDIT program is the same whether the user is at the system console or at a remote terminal.

C. Commands

2.12 The text editor is instructed as to what function it is to perform via entry of command mnemonics. When entering these commands via the keyboard, they should begin immediately after the text editor prompter

(EDT?). The text editor will not begin executing the command until a carriage return is entered. If a mistake is made in entering characters, the user can correct the mistake via the following operations:

KEYBOARD OPERATION	ACTION
Shift O	Deletes the last character entered. Two of these operations will delete the last two characters, etc
RUBOUT and RETURN	Deletes the entire line being entered

If an illegal command or operation is specified, an error message will appear. Error messages are listed in Part 2D.

2.13 Most of the command mnemonics can be abbreviated to a single character. The exact abbreviation will be given with each command description. Key words may be inserted as parameters in certain commands, but they ***cannot be abbreviated.*** Permissible key words are presented in Table A. Certain commands may require the use of a peripheral device. A description of the peripheral devices is shown in Table B.

2.14 Many of the text editor commands perform a function over some range specified by the operator or in some cases to some default range. By default, we mean that if a value is not entered by the operator, the text editor will assume some predictable value. Range parameters may be specified as line number or character strings. A range specified by character strings is mainly intended for use in editing data base records since each record is identified by type with special characters. For example, this would provide for editing over a set of records associated with a particular trunk group.

2.15 Generally, if a character string appears to the left of the command mnemonic as part of a range entry, it should be enclosed in quotes ("string"). Conversely, if the character string appears to the right of the command mnemonic, quotes need not be entered.

2.16 In various commands, some parameters are optional and others are required inputs. Special designations are provided in the following command descriptions to distinguish between the two types. ***Optional parameters are enclosed in left and right brackets ([and]), while required parameters are enclosed in left and right arrows (< and >).***

In the format of the following command mnemonics, the delta character (Δ) is used to indicate a space character (space bar). The actual space character will be used in examples.

APPEND Command, Abbreviation A

Append lines of a disc file to the work-space file.

Format: *[n1;n2,n3]*Δ**APPEND**Δ*<file name>*

Defaults: *n1;n2* equals all of file named; *n3* equals current line.

Where: *n1* and *n2* specify the range of lines in the specified *<file name>* to be appended in the work-space file after line *n3.*

Example 1: **1;1Ø,1 APPEND TRTEST**—Append lines 1 through 1Ø of TRTEST after line 1 of the work-space file.

Example 2: **1Ø APPEND TASC** —Append all of file TASC to work-space file beginning after line 1Ø.

Example 3: **APPEND TASC**—Append all of file TASC to work-space file beginning after the current line.

COL Command, Abbreviation K

Output a line of numbers which indicate line columns.

Format: **COL**Δ*[device]*

Default: [*device]* **TE**

Where: *device* is a key word which represents the output device line printer **(LP)** or user terminal **(TE)**.

Example 1: **COL LP**—Output column designators to line printer.

Example 2: **COL**—Output column designators to user terminal.

Comments: This command is intended to be used with keyboard entry of data which must be positioned in certain columns. The output will be similar to the following:

```
0         1         2         3
12345678901234567890123456789012345. . . .
```

COPY Command, Abbreviation C

Copy lines from one part of work-space file to another part.

Format: *[n1;n2,n3]*Δ**COPY**

Defaults: *n1,n2,n3* equal to current line.

Where: *n1* and *n2* specify the range of lines to be copied to immediately after line *n3; n1,n2,* and *n3* will default to the current line if not specified. Unlike the **MOVE** command, the original lines *n1* and *n2* are not deleted.

Example 1: ***;114,150 COPY**—Copy range of lines from current line to 114 to follow line 150.

Example 2: ***,60 COPY**—Copy current line to follow line 60.

DELETE Command, Abbreviation D

Delete lines from work-space file.

Format: *[n1;n2]*Δ **DELETE**Δ[*string*]

Defaults: *n1,n2* = current line.

Where: If no string is specified, then *n1* and *n2* specify the range of lines to be deleted. If *n1* and *n2* are not specified, the current line is deleted. If only *n1* is specified, only that line is deleted.

Example 1: **110;100 DELETE**—Delete range of lines 10 through 100.

Example 2: **56 DELETE**—Delete line 56.

Example 3: **DELETE**—Delete current line.

Comments: If a *string* is specified, then *n1* and *n2* specify the range over which only those lines containing the string specified are deleted. This is intended to be used for deleting lines containing certain information from a file.

Example 4: **1;# DELETE /CT**—Delete all lines containing the string **/CT**.

Example 5: **1;# DELETE ABC(15;18)**—Delete all lines with string ABC in columns 15 through 18.

DL Command, No Abbreviation

Give a directory list of all user files.

Format: **DL**Δ*[device]*

Default: *[device]* equals user terminal.

Comments: This command may be used to obtain a list of all user files. They may be displayed on the user terminal or on the line printer (device=**LP**) if the user is at the system console.

EDIT Command, Abbreviation E

Edit a line in the text file.

Format: *[n]*Δ**EDIT**

Default: *n* equals current line.

Where: *n* is line number of line to be edited.

Comments: This command is used to edit a single line by use of subcommands. When **EDIT** is executed, the line to be edited is displayed and the prompt **?** appears. **EDIT** at this point is awaiting the input of one of the following subcommands:

I<S1>—Insert the character string S1 in the line at the point immediately in front of the character above the I.

D<S1>Delete the characters in the line starting immediately above the D, deleting one character for each character in the string S1.

R<S1>—Replace the characters in the line starting immediately above the R on a one-for-one basis with the character string S1.

.—an entry consisting of a single period in the first column exits the subcommand mode.

Example:

```
14 EDIT
   /ADATASSEØXYZ1

?          ILNG

   /ADATLNGASSEØXYZ1

?                  DLLL

   /ADATLNGASSEØ1

?   RCT

   /CTATLNGASSEØ1

?.
```

Note: If a blank was entered after the subcommand, the blank would be considered part of the ASCII string.

END Command, No Abbreviation

Exit the text editor.

Format: **END**

Comments: This command releases the work-space file for other CAROT uses. If the user has not saved his data by use of the **SAVE**, or **WRITE** commands, then all editing will be lost.

The **END** command also closes all opened files, outputs end-of-file markers to the magnetic tape and paper tape punch and a top of form to the line printer, if they were used with the **WRITE** command.

FIND Command, Abbreviation F

Search the work-space file and display the next line containing the specified string.

Format: *[n1;n2]*Δ**FIND**Δ*<character string>*

Defaults: *n1* equals current line; *n2* equals last line.

Where: *n1* and *n2* specify the range of lines to be searched for the specified string, and *<character string>* is a string of characters to be searched for.

Example 1: **1Ø;15Ø FIND /IDATLNGAEP76A**—Find the specified string between lines 1Ø and 15Ø.

Example 2: **FIND IETGTC**—Find the specified string between the current line and last line.

Comments: Once the search string is set up, it is stored intact until another **FIND** command with a new string is entered. This provides the operator the capability to simply use the format *[n1;n2]* **F** (or just **F**) without having to reenter the search string if the previous **FIND** command string is to be used.

HELP Command, Abbreviation H

Obtain a description of a command syntax.

Format: **HELP**Δ*<mnemonic>Δ[device]*

Defaults: *<mnemonics>* equals ALL, *[device]* equals **TE.**

Where: *<mnemonic>* is a single character or full word representing the command for which a description is desired. Enter **ALL** for mnemonic if a description of all the commands is desired. *[device]* is a key word which represents the output device **LP** or **TE.**

Example 1: **HELP FIND**—Output description of the **FIND** command on user terminal.

Example 2: **HELP ALL,LP**—Output a description of all commands on the line printer.

ILINE, Command, Abbreviation I

Insert lines in the work-space file.

Format: *[n]*Δ**ILINE**

Defaults: *n* = current line.

Where: *[n]* indicates the line designation after which lines are to be inserted. Terminate with a line containing a single period (.).

Example (with prompts in bold type):

EDT?100 ILINE

?REATLNGATH63AØ126.21—Line of text to be inserted after line 1ØØ

?ODATLNGATH63ASXS—Line of text to be inserted after line 1ØØ

?.—indicates to terminate

EDT?

Comments: This command indicates that the next lines typed are to be inserted immediately after line *n1*. The prompt changes to a single question mark (**?**) for the lines to be inserted. The user may enter as many lines as is desired. To exit from the line insertion, enter a line containing a single period (.). The prompt will then revert back to **EDT?**.

The **ILINE** command can be used to load data into the work-space file from the keyboard. This is normally the case when inputting CAROT/RPG source programs. Type **ILINE** and return to start the file.

LOAD Command, Abbreviation L

Load data into the work-space file.

Format: **LOAD**Δ*<device or file name>*

Where: *<device> is a key word representing a peripheral device (PT or MT), and <file name> is a name of a disc file.*

Comments: After the data is loaded, the first line will be displayed and the prompt **EDT?** will indicate that the program is ready to accept the next command. If the specified file cannot be found, an error message will be outputted. Loading a disc file into the work space file does not affect the original file.

MOVE Command, Abbreviation M

Move lines from one part of work-space file to another part.

Format: *[n1;n2,n3]*Δ**MOVE**

Defaults: *n1,n2,n3,* equal to current line.

Where: *n1* and *n2* specify the range of lines to be moved to immediately follow line *n3.* The original lines *n1* and *n2* are deleted.

PURGE Command, Abbreviation P

Purge a previously created file.

Format: **PURGE**Δ*<file name>*

Where: *<file name>* is the name of the file to be deleted.

Example 1: **PURGE TRKTST**

Example 2: **PURGE FILEA**

If the file to be purged does not exist, an error message will be given.

REPLACE Command, Abbreviation R

Replace one or more lines with other lines.

Format: *[n1;n2]*Δ**REPLACE**

Defaults: *n1* and *n2* equal current line.

Where: *n1* and *n2* specify the range of lines to be deleted and replaced with the following lines. As many replacement lines as desired may be entered. Terminate the entry of lines with a line containing a single period (.).

Example (with prompts in bold type):

EDIT?1Ø;15 REPLACE

?THE DOG HAS FLEAS.—Lines of text to be put in place of lines 10 to 15.

?THE CAT HAS FLEAS.—Lines of text to be put in place of lines 10 to 15.

?.—Indicates to terminate.

EDT?

Comments: The **REPLACE** command is essentially a combination of the delete (**D**) and insert line (**I**) commands.

SAVE Command, Abbreviation S

Save work-space file in a user file.

Format: *[n1;n2]*Δ**SAVE**Δ<*file name*>Δ*[***UP**]

Defaults: All lines, no **[UP]**.

Where: *n1* and *n2* specify the range of lines in the work-space file to be saved in the file. <*file name*> is the name of the file in which the data is to be stored. <*file name*> may be left blank if the desired file is the same as the one in a previous save. If an asterisk (*) is entered as the <*file name*>, the data will replace the original file entered with the **LOAD** command. Do not use MT, PT, or LT as a <*file name*>. **UP** is an optional parameter which indicates that the file is to be used during the next update.

Example 1: **1Ø;1ØØ SAVE RFILE**—Save lines 1Ø through 1ØØ in file RFILE and use for update.

Example 2: **SAVE TEMP1**—Save entire contents of work-space file in file TEMP1.

Example 3: **114 SAVE TEMP1**—Save line 114 in TEMP1.

Example 4: **SAVE * UP**—Replace original file and flag it for update.

Comments: *n1* and *n2* specify the range of lines in the work-space file to be saved in the specified file. If *n1* and *n2* are not given, all lines of the work-space file will be saved. If only *n1* is given, only that line will be saved. On the first save to a particular file, the file will be opened. Subsequent saves to the same file will be written at the end of the file until the editor is terminated. When the editor is terminated, an end-of-file mark is written at the end of the file. Attempts to save data in an already existing file which was not created during the current run of the editor will cause an error message. If it is desired to overwrite a previously used file, first purge the old file using the **PURGE** command.

The optional update flag, **UP**, indicates that the file should be used as input data for the next run of the update cycle. The data will only be used for one update, but the file will not be purged. This will allow the user to make changes to the file using the editor and to use the new data for the next update. This feature was incorporated for instances in which there was an error in the data which caused the update data to be rejected. The maximum number of files which can be flaged for update is 31. Any attempt to set more than 31 files will result in an error 12 (cannot set update flag).

SET Command, Abbreviation SE

Four Forms

Set the maximum length of a line. ›

Format: **SET:LENGTH=**<*n*>

Default: <*n*> = 72.

Where: <*n*> is the maximum length of each line to be displayed.

Example: **SET:LENGTH=40**—This will truncate all lines after the 40th character.

> ***Caution: Loading of files greater than 72 characters per line will result in truncated records. This is particularly true for ROTL records (RO). Also, some RPG/CAROT programs may be damaged. Be sure to set LENGTH to greater than the number of characters per line that is required.***

Set the don't care character.

Format: **SET:DC=**<*character*>

Default: <*character*> = @

Where: <*character*> is the don't care character to be used in string parameters.

Example: **SE:DC=/**

Set the quote character.

Format: **SET:QUOTE=**<*character*>

Default: <*character*> = "

Where: <*character*> is any ASCII character to be used as the quote to enclose character strings.

Example: **SE:QUOTE=:**

Set a character string.

Format: **SET:**<*label=character string*>

Default: All null and zero length.

Where: <*label*> is one of the three string labels $1, $2, or $3, and <*character string*> is any group of ASCII characters.

Example 1: **SET:$3=ATLNGAEP76A**

Example 2: **SET:$1=ABCD(13;16)**

SHOW Command, Abbreviation SH

Show the state of parameters set by the **SET** command.

Formats: **SHOW LENGTH**—Show maximum line length.
SHOW DC—Show the don't care character.
SHOW $*n*—Show the contents of the character string.
SHOW LAST—Show the last line number in the work-space file.
SHOW QUOTE—Show quote character.
SHOW ALL—Show the state of all of the above.

TAB Command, Abbreviation T

Specify tab character and tab points.

Format: **TAB**Δ*[c],[n1,n2,n3.....]*

Defaults: *[c]*=' (single quote), *[n1,n2,n3....]* clear (or **TAB ',1,10,20)**

Where: *[c]* is a single nonnumeric character which if entered in a character string results in cursor movement to next tab point. *[n1,n2,n3....]* are numerals indicating column positions to be used as tab points.

Example 1: **TAB %,7,12,40**—Change tab character to % and specify three tab points.

Example 2: **TAB 8,15**—Specify two tab points and continue with present tab character.

Comments: When the tab character is entered with any of the other commands, the cursor moves to the tab point filling in with don't care characters. For example, assume that the tab character is %, the don't care character is @, and a tab point is set at column 1Ø. Entry of the string ABCD%EF would result in a string of ABCD@@@@@@EF.

To understand this concept further, input the following messages when using the EDIT program.

```
EDIT?TAB %,10

EDIT?SET:$1ABCD%EF

EDIT?SHOW $1
```

The resulting printout will look like:

```
$1 ABCD@@@@@@EF
```

WRITE Command, Abbreviation W

List lines of text in work-space file on user terminal or printer, or output to magnetic or paper tape.

Format: *<n1;n2>*Δ**WRITE**Δ*[device,*[**UN/LN**]]

Default: User terminal with line number for **LP** or **TE** and no line number for **MT** or **PT** is entered.

Where: *n1* and *n2* specify the range of text to be outputted. If only *n1* is entered, only that line is outputted. *device* is a key word representing an output device (Table C). **UN** is entered if one wants an unnumbered (no line numbers) output. **LN** is entered for a numbered output.

Example 1: **14;1ØØ WRITE**—Lists lines 14 through 1ØØ on user terminal with line numbers.

Example 2: **36 WRITE LP,UN**—Output line 36 on line printer without line numbers.

Example 3: **Ø;# WRITE MT**—Output entire file to magnetic tape without line numbers.

Comments: When magnetic tape (MT) is specified as the output device, records are recorded in an unlabeled, unblocked format. Line numbers are not outputted, unless specified, when outputting to magnetic or paper tape. Output to MT, PT or LP is only available if EDIT is being run from CAROT console.

XCHANGE Command, Abbreviation X

Exchange one character string for another.

Format: *[n1;n2]*Δ**XCHANGE**Δ*<search string>; <replacement string>*

Defaults: *n1* equals current line; *n2* equals last line.

Where: *n1* and *n2* specify a range of lines to search in the work-space file, *<search string>* is the ASCII string to search for within the range of lines, and *<replacement string>* is the ASCII string which is to replace all occurrences of the search string.

Example 1: **1Ø;1ØØ XCHANGE** $1;$2—Search lines 1Ø through 1ØØ for the string represented by $1 and replace with the string represented by $2.

Example 2: **XCHANGE ATAG;ATAL**—Search work-space file from current line to end for string ATAG and replace all occurrences with ATAL.

Comments: In this command, the column number qualifiers are meaningful only to the search string. It should be noted that line numbers can be character strings. For example, by specifying *n1* as a facility "ID" and *n2* as "/TF", it would be possible to change trunk transmission parameters (such as EML) for all of the trunks within a single-facility group in the data base.

TABLE A

PERMISSIBLE KEY WORDS IN EDITOR COMMANDS

KEY WORD	MEANING
ALL	All parameter
LAST or #	Last line in file
*	Current line
*+5	Fifth line after current line
*-5	Fifth line before current line
$1 — $3	Labels used represent a character string

TABLE B

KEY WORDS WHICH REPRESENT PERIPHERAL DEVICES

KEY WORD	DEVICE
TE	User terminal
PT	Paper tape reader (input) or punch (output)
MT	Magnetic tape
LP	Line printer

TABLE C

TEXT EDITOR ERROR NUMBERS AND MESSAGES

ERROR NUMBER	MESSAGE
1	FILE NOT FOUND
2	LIST INTERRUPTED
3	SYMBOL TABLE OVERFLOW
4	INVALID LINE NUMBER
5	INVALID SYNTAX
6	INVALID KEYWORD
7	INVALID KEYWORD VALUE
8	INVALID KEYWORD LENGTH
9	FILE ALREADY EXISTS
10	NO SPACE ON DISC
11	INVALID FILE NAME
12	CAN'T SET UPDATE FLAG
13	UPDATE – PLEASE END EDIT

D. Error Messages

2.17 When an error occurs, the text editor will output a message of the form *nn text* where *nn* is an error number and *text* is a description of the error associated with a particular illegal operation. A list of error messages associated with the text editor is provided in Table C.

3. LIST UPDATE FILES

3.01 The program LFILE provides a means of listing the names of files which are currently scheduled to be used for input at the next update cycle. This list may be output to the user terminal or on the system line printer if the user is at the system console.

3.02 To run the program, the user types the following:

RUN:LFILE. If the user is at a remote terminal, the program will respond with the current list of files or a message indicating that no files are scheduled for update. If the user is located at the system console, the program will respond with the following message:

SPECIFY IF FILE LIST IS TO BE OUTPUT ON YOUR TERMINAL BY TYPING "TE" OR ON THE LINE PRINTER BY TYPING "LP".

If there are no files scheduled for update, the following message will be printed:

NO FILES SCHEDULED FOR UPDATE

4. APPLICATION EXAMPLES

4.01 Up to this point in this section, the *theory of use* of the commands and parameters have been explained. Due to the nature of the EDIT program, it is very difficult to discuss the *application of use* of the commands. This is because there are many possisble variations of the commands with parameters; plus the fact that the problem (ie, what needs to be done) cannot be strictly defined. In an attempt to fill this gap, the following is a collection of examples which demonstrate the use of the EDIT program. Study these examples to gain further insight into the *applications* of the commands.

4.02 All of these actual examples were generated from a remote-user location and show the commands and the resulting printout. Almost all of the examples used in this part use a disc file called NAME1. The NAME1 disc file was generated using the SELEC program (Section 190-102-206). An example of the SELEC commands used to generate the file is shown in the following:

```
? RUN:SELECT
SEL?SET:ROTL=NDADFLGG65A0
SEL?SET:OFFZ=0JUSFLTL03T
SEL?SET:OUTPUT=NAME1
SEL?FIND:TG
SEL?END
?
```

Keep in mind that some of the examples presented are for example purposes only and do not reflect normal usage. This unusual usage allows the user more visibility into the operation of the EDIT commands.

A. LOAD and WRITE Commands

4.03 ***Example 1:*** Load into the editor work-space file a disc file called NAME1.

```
? RUN:EDIT
EDT?LOAD NAME1
   1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
EDT?
```

Notice on the printout that after the file has been loaded, the current line (in this case, the first line) will be printed out on the terminal. This is also true for many of the other commands in this part.

4.04 ***Example 2:*** Using the NAME1 file loaded in example 1, display the information on the terminal.

```
EDT?0;# WRITE
   1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   3  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
   5  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   7  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
   9  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
EDT?
```

4.05 ***Example 3:*** List line 5.

```
EDT?5 WRITE
   5  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
EDT?
```

4.06 ***Example 4:*** List all lines to the terminal without any line numbers.

```
EDT?0;# WRITE TE,UN
/RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
/YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
/YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
/YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
/YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
/YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
/YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
/YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
   9  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
EDT?
```

4.07 ***Example 5:*** List lines 5 through 8 to the terminal.

```
EDT?5;8 WRITE
   5  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   7  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
EDT?
```

4.08 ***Example 6:*** List all lines but use BREAK key to terminate printout.

```
EDT?1;# WRITE
    1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
*** BREAK ***
GO
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
    5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A          TH@
*** BREAK ***
CAT
INVALID RESPONSE - STOP OR GO?

*** BREAK ***
STOP
EDT?
```

B. FIND Command

4.09 ***Example 7:*** Find all lines that contain the character string **YBN.**

```
EDT?1;# F YBN
    3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?FIND
    5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?FIND
    7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?FIND
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

4.10 ***Example 8:*** Find all lines that contain the character string **03T A.**

```
EDT?1;# FIND 03T A
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
EDT?FIND
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
EDT?FIND
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
EDT?FIND
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
EDT?
```

4.11 ***Example 9:*** Find all lines that contain the character string **Ø3T A** using character string label **$1**.

```
EDT?SET:$1=03T A
    1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
EDT?FIND,$1
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
EDT?FIND
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
EDT?FIND
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
EDT?FIND
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
EDT?
```

C. INSERT Command

4.12 ***Example 10:*** Insert additional lines into the work-space file after line 4.

```
EDT?4 ILINE
      ?***** THE OLD DOG HAS FLEAS. *****
      ?***** BUT HE DOES NOT CARE.  *****
      ?.
    5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?1;# WRITE
    1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
.   1  ***** THE OLD DOG HAS FLEAS. *****
.   2  ***** BUT HE DOES NOT CARE.  *****
    5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

D. DELETE Command

4.13 ***Example 11:*** Delete lines 4.1 and 4.2 which were inserted in the last example.

```
EDT?4.1;4.2 DELETE
    5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?1;# WRITE
    1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
    5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

4.14 ***Example 12:*** Delete all lines starting at line 7 to the end of the file.

```
EDT?7;9 DELETE
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
EDT?1;# WRITE
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   3  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
   5  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
EDT?
```

4.15 ***Example 13:*** Delete line 3.

```
EDT?3 DELETE
   4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
EDT?1;# WRITE
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
   5  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   7  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
   9  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
EDT?
```

4.16 ***Example 14:*** Delete all lines that contain the character string **/YGD**.

```
EDT?1;# DELETE /YGD
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
EDT?1;# WRITE
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
   3  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   5  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   7  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   9  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
EDT?
```

4.17 ***Example 15:*** Delete all lines that contain the character string 65 in columns 13 and 14.

```
EDT?1;# DELETE 65(13;14)
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
EDT?1;# WRITE
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
   6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
EDT?
```

E. COPY Command

4.18 ***Example 16:*** Copy lines 1 through 3 to begin after line 7.

```
EDT?1;3,7 COPY
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
EDT?1;#,WRITE
   1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
   4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
   5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
   6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
.  1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
.  2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
.  3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
   9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
   1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
EDT?
```

4.19 ***Example 17:*** Copy current line (in this case line 8) to begin after line 1.

```
EDT?*,1 COPY
   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
EDT?1;# WRITE
   1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
.  1  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
   4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
   5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
   6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
   9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

4.20 ***Example 18:*** Copy all lines and place after the last line.

```
EDT?1;#,# COPY
   1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
EDT?1;# WRITE
   1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   3  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
   5  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   7  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
   9  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
.  1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
.  2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
.  3  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
.  4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
.  5  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
.  6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
.  7  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
.  8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
.  9  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
EDT?
```

F. MOVE Command

4.21 Move lines 6 and 7 to fall before line 1.

```
EDT?6;7,0 MOVE
   1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
EDT?0;# WRITE
.  1  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
.  2  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   3  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
   5  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
   9  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
EDT?
```

4.22 Move lines 1 through 3 to fall after line 8.

```
EDT?1;3,8 MOVE
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?1;# WRITE
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
    5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
.   1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
.   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
.   3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

G. REPLACE Command

4.23 ***Example 19:*** Replace line 5 with new lines.

```
EDT?5;5 REPLACE
      ?RATS EAT CHEESE
      ?CATS EAT RATS
      ?.
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
EDT?1;# WRITE
    1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
    5  RATS EAT CHEESE
.   1  CATS EAT RATS
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

4.24 ***Example 20:*** Replace lines 3 through 7 with new lines.

```
EDT?3;7 REPLACE
      ?IF YOU CAN'T SAY SOMETHING GOOD ABOUT YOUR FRIENDS--
      ?THEN YOU HAVE THE WRONG FRIENDS !!
      ?.
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
EDT?1;# WRITE
    1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    7  IF YOU CAN'T SAY SOMETHING GOOD ABOUT YOUR FRIENDS--
.   1  THEN YOU HAVE THE WRONG FRIENDS !!
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

H. EDIT Command

4.25 ***Example 21:*** This example shows the insert, replace, and delete command usage. The first part inserts 123456789 character string into line 7 between **/YBNDA** and **DFLGGE6---**. The second part replaces 123456789 with **XXXXXXXXX**. Finally, the **XXXXXXXXX** is deleted returning the line to its original state.

```
EDT?7 EDIT
   7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
     ?       I123456789
   7  /YBNDA123456789DFLGGE65NDADFLGG65A00JUSFLTL03T

     ?       RXXXXXXXXX
   7  /YBNDAXXXXXXXXXDFLGGE65NDADFLGG65A00JUSFLTL03T

     ?       D**********
   7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T

     ?.
   7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T

EDT?
```

I. XCHANGE Command

4.26 ***Example 22:*** Search all lines for the character string **FL** and exchange it with a ++ character string.

```
EDT?1;# XCHANGE FL;++
   1  /RONDAD++GG65A05XB203056510868  213011              NDAD++GG65A0000 000000
   2  /YGDF54CA-ANI     NDAD++GG65A M- OJUS++TL03T A
   3  /YBNDAD++GGE65NDAD++GG65A00JUS++TL03T
   4  /YGDF54CA-CR      NDAD++GG65A M- OJUS++TL03T A
   5  /YBNDAD++GGE65NDAD++GG65A00JUS++TL03T
   6  /YGDF54DA-ANI     NDAD++GG65A M- OJUS++TL03T A
   7  /YBNDAD++GGE65NDAD++GG65A00JUS++TL03T
   8  /YGDF54SP-COMB    NDAD++GG65A M- OJUS++TL03T A
   9  /YBNDAD++GGE65NDAD++GG65A00JUS++TL03T
   1  /RONDAD++GG65A05XB203056510868  213011              NDAD++GG65A0000 000000
EDT?
```

4.27 ***Example 23:*** Search lines 2 through 8 for the character string **OJUS** and exchange it with a **ZZZZ** character string.

```
EDT?2;8 XCHANGE OJUS;ZZZZ
   2  /YGDF54CA-ANI     NDADFLGG65A M- ZZZZFLTL03T A
   3  /YBNDADFLGGE65NDADFLGG65A0ZZZZFLTL03T
   4  /YGDF54CA-CR      NDADFLGG65A M- ZZZZFLTL03T A
   5  /YBNDADFLGGE65NDADFLGG65A0ZZZZFLTL03T
   6  /YGDF54DA-ANI     NDADFLGG65A M- ZZZZFLTL03T A
   7  /YBNDADFLGGE65NDADFLGG65A0ZZZZFLTL03T
   8  /YGDF54SP-COMB    NDADFLGG65A M- ZZZZFLTL03T A
   9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

4.28 ***Example 24:*** Search for first line containing character string **ANI** and exchange **A** for – (minus sign) until the character string **COMB** is found.

```
EDT?"ANI";"COMB" XCHANGE A;-
   2  /YGDF54C---NI     ND-DFLGG65- M- OJUSFLTL03T -
   3  /YBND-DFLGGE65ND-DFLGG65-0OJUSFLTL03T
   4  /YGDF54C--CR      ND-DFLGG65- M- OJUSFLTL03T -
   5  /YBND-DFLGGE65ND-DFLGG65-0OJUSFLTL03T
   6  /YGDF54D---NI     ND-DFLGG65- M- OJUSFLTL03T -
   7  /YBND-DFLGGE65ND-DFLGG65-0OJUSFLTL03T
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
EDT?1;# WRITE
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
   2  /YGDF54C---NI     ND-DFLGG65- M- OJUSFLTL03T -
   3  /YBND-DFLGGE65ND-DFLGG65-0OJUSFLTL03T
   4  /YGDF54C--CR      ND-DFLGG65- M- OJUSFLTL03T -
   5  /YBND-DFLGGE65ND-DFLGG65-0OJUSFLTL03T
   6  /YGDF54D---NI     ND-DFLGG65- M- OJUSFLTL03T -
   7  /YBND-DFLGGE65ND-DFLGG65-0OJUSFLTL03T
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
   9  /YBNDADFLGGE65NDADFLGG65A0OJUSFLTL03T
EDT?
```

4.29 ***Example 25:*** Set $1 equal to character string OJUS and $2 equal to ABCDE. Search all lines for character string equal to $1 and exchange it for character string equal to $2.

```
EDT?SET:$1=OJUS
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
EDT?SET:$2=>ABCDE<
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
EDT?XCHANGE $1;$2
   2  /YGDF54CA-ANI     NDADFLGG65A M- >ABCDE<FLTL03T A

   3  /YBNDADFLGGE65NDADFLGG65A0>ABCDE<FLTL03T

   4  /YGDF54CA-CR      NDADFLGG65A M- >ABCDE<FLTL03T A

   5  /YBNDADFLGGE65NDADFLGG65A0>ABCDE<FLTL03T

   6  /YGDF54DA-ANI     NDADFLGG65A M- >ABCDE<FLTL03T A

   7  /YBNDADFLGGE65NDADFLGG65A0>ABCDE<FLTL03T

   8  /YGDF54SP-COMB    NDADFLGG65A M- >ABCDE<FLTL03T A

   9  /YBNDADFLGGE65NDADFLGG65A0>ABCDE<FLTL03T

   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
EDT?
```

4.30 ***Example 26:*** Search all lines for the character string **DAD**, in columns 5, 6, and 7, and exchange it for character string **XXX**. (Notice in the printout that only the **DAD** in columns 5, 6, and 7 have been exchanged.)

```
EDT?1;# XCHANGE DAD(5;7);XXX
   1  /RONXXXFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
   3  /YBNXXXFLGGE65NDADFLGG65A00JUSFLTL03T
   5  /YBNXXXFLGGE65NDADFLGG65A00JUSFLTL03T
   7  /YBNXXXFLGGE65NDADFLGG65A00JUSFLTL03T
   9  /YBNXXXFLGGE65NDADFLGG65A00JUSFLTL03T
   1  /RONXXXFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
EDT?1;# WRITE
   1  /RONXXXFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   3  /YBNXXXFLGGE65NDADFLGG65A00JUSFLTL03T
   4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
   5  /YBNXXXFLGGE65NDADFLGG65A00JUSFLTL03T
   6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
   7  /YBNXXXFLGGE65NDADFLGG65A00JUSFLTL03T
   8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
   9  /YBNXXXFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

4.31 ***Example 27:*** Seach all lines for the character string **DAD**, after column 18, and exchange it for character string **ZZZZ**.

```
EDT?1;# XCHANGE DAD(18);ZZZZ
   1  /RONDADFLGG65A05XB203056510868  213011            NZZZZFLGG65A0000 00000

   2  /YGDF54CA-ANI     NZZZZFLGG65A M- OJUSFLTL03T A

   4  /YGDF54CA-CR      NZZZZFLGG65A M- OJUSFLTL03T A

   6  /YGDF54DA-ANI     NZZZZFLGG65A M- OJUSFLTL03T A

   8  /YGDF54SP-COMB    NZZZZFLGG65A M- OJUSFLTL03T A

   1  /RONDADFLGG65A05XB203056510868  213011            NZZZZFLGG65A0000 00000
EDT?1;# WRITE
   1  /RONDADFLGG65A05XB203056510868  213011            NZZZZFLGG65A0000 00000
   2  /YGDF54CA-ANI     NZZZZFLGG65A M- OJUSFLTL03T A
   3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
   4  /YGDF54CA-CR      NZZZZFLGG65A M- OJUSFLTL03T A
   5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
   6  /YGDF54DA-ANI     NZZZZFLGG65A M- OJUSFLTL03T A
   7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
   8  /YGDF54SP-COMB    NZZZZFLGG65A M- OJUSFLTL03T A
   9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

4.32 ***Example 28:*** Using the exchange command, insert the character string **123** after every occurrence of the character string **FLG**.

```
EDT?1;# XCHANGE FLG;FLG123
    1  /RONDADFLG123G65A05XB203056510868  213011              NDADFLG123G65A0000

    2  /YGDF54CA-ANI     NDADFLG123G65A M- OJUSFLTL03T A

    3  /YBNDADFLG123GE65NDADFLG123G65A00JUSFLTL03T

    4  /YGDF54CA-CR      NDADFLG123G65A M- OJUSFLTL03T A

    5  /YBNDADFLG123GE65NDADFLG123G65A00JUSFLTL03T

    6  /YGDF54DA-ANI     NDADFLG123G65A M- OJUSFLTL03T A

    7  /YBNDADFLG123GE65NDADFLG123G65A00JUSFLTL03T

    8  /YGDF54SP-COMB    NDADFLG123G65A M- OJUSFLTL03T A

    9  /YBNDADFLG123GE65NDADFLG123G65A00JUSFLTL03T

    1  /RONDADFLG123G65A05XB203056510868  213011              NDADFLG123G65A0000
EDT?
```

4.33 ***Example 29:*** Using the exchange command, insert the character string +++ after the character string **DAD** in columns 5, 6, and 7 only.

```
EDT?XCHANGE DAD(5;7);DAD+++
    1  /RONDAD+++FLGG65A05XB203056510868  213011              NDADFLGG65A0000 000

    3  /YBNDAD+++FLGGE65NDADFLGG65A00JUSFLTL03T

    5  /YBNDAD+++FLGGE65NDADFLGG65A00JUSFLTL03T

    7  /YBNDAD+++FLGGE65NDADFLGG65A00JUSFLTL03T

    9  /YBNDAD+++FLGGE65NDADFLGG65A00JUSFLTL03T

    1  /RONDAD+++FLGG65A05XB203056510868  213011              NDADFLGG65A0000 000
EDT?1;# WRITE
    1  /RONDAD+++FLGG65A05XB203056510868  213011              NDADFLGG65A0000 000
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    3  /YBNDAD+++FLGGE65NDADFLGG65A00JUSFLTL03T
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
    5  /YBNDAD+++FLGGE65NDADFLGG65A00JUSFLTL03T
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    7  /YBNDAD+++FLGGE65NDADFLGG65A00JUSFLTL03T
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
    9  /YBNDAD+++FLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
?
```

J. APPEND Command

4.34 ***Example 30:*** Append to the work space file, after the current line, the file named **FILEC.**

```
EDT?1;# WRITE
    1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
    5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?APPEND FILEC
    1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
EDT?1;# WRITE
    1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
    5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
.   1  AAAAAAAAAAAAAAAAAAAAAAAAA
.   2  BBBBBBBBBBBBBBBBBBBBBBB
.   3  CCCCCCCCCCCCCCCCCCCCCCC
EDT?
```

Note: Notice that **FILEC** was appended to fall after the current line. In this case, it was the last line because of the previous **1;#,WRITE** command. In the following example, the current line is line 7.

4.35 ***Example 31:*** Append to the work space file, after the current line, the file named **FILEC.**

```
EDT?APPEND FILEC
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
EDT?1;# WRITE
    1  /RONDADFLGG65A05XB203056510868  213011              NDADFLGG65A0000 000000
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    3  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
    5  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
    7  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
.   1  AAAAAAAAAAAAAAAAAAAAAAAAA
.   2  BBBBBBBBBBBBBBBBBBBBBBB
.   3  CCCCCCCCCCCCCCCCCCCCCCC
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

K. SAVE Command

4.36 ***Example 32:*** Save the work space file to disc using the name TEMP1.

```
EDT?SAVE TEMP1
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
EDT?
```

L. Purge Command

4.37 ***Example 33:*** Purge a file on disc called TEMP1.

```
EDT?PURGE,TEMP1
EDT?
```

4.38 ***Example 34:*** Purge a file on disc called TGFILE.

```
EDT?PURGE TGFILE
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
EDT?
```

M. COL Command

4.39 ***Example 35:*** Print out column number header.

```
EDT?COL
            1         2         3         4         5         6         7
   12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
   1  /RONDADFLGG65A05XB203056510868  213011            NDADFLGG65A0000 000000
```

N. SET and SHOW Commands

4.40 ***Example 36:*** The following shows several examples of the **SET** and **SHOW** commands.

```
EDT?SET:$1=ABCDEF
EDT?SET:$2=WXYZ
EDT?SET:LENGTH=70
EDT?SET:DC=%
EDT?SHOW:ALL
 LENGTH =    70
 DC =   %
 $1 = ABCDEF(   0; 72)
 $2 = WXYZ(   0; 72)
 $3 =
 LAST =     0.     0
 TABS =   '
EDT?SET:$1=ABC(20;22)
EDT?SHOW:ALL
 LENGTH =    70
 DC =   %
 $1 = ABC( 20; 22)
 $2 = WXYZ(   0; 72)
 $3 =
 LAST =     0.     0
 TABS =   '
EDT?
```

O. DL Command

4.41 ***Example 37:*** Print out all the disc files accessible by the EDIT program.

```
EDT?DL
  COFCFI
  FIBUSY
  DIST1
  DIST2
  DIST3
  DIST4
  FIMNGS
  @AUDIT
  @BLOK
  @CHG
  @COLO
  @COPY
  @COTC1
  NAME1
  FILEC
  FILEBC
  NAME2
  FILEA
  FILEB
  THIS1
  KEN
  TGFILE
  @BRHM
  @BUSY
  '$256A
  '$256B
   2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
EDT?
```

P. TAB Command

4.42 The use of the **TAB** command can provide some interesting possibilities when used in conjunction with the other EDIT commands. The examples below assume the TAB character is a single quote (') and the don't care character is an 'at sign' (@).

4.43 ***Example 38:*** Set up a $1 string which uses the **TAB** character to fill in the don't care columns up to but not including column 20.

```
EDT?TAB 20
   9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?SET:$1=WXYZ'Z
   9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?SHOW $1
 $1 = WXYZ@@@@@@@@@@@@@@@@@@Z(  0: 72)
   9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?
```

4.44 ***Example 39:*** Use the **TAB** character to go to column 11 and XCHANGE the next four characters with XYZA.

```
EDT?TAB 11
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?0;# XCHANGE "/YG'@@@@";"/YG'XYZA"
    2  /YGDF54CA-XYZA    NDADFLGG65A M- OJUSFLTL03T A
    4  /YGDF54CA-XYZA    NDADFLGG65A M- OJUSFLTL03T A
    6  /YGDF54DA-XYZA    NDADFLGG65A M- OJUSFLTL03T A
    8  /YGDF54SP-XYZA    NDADFLGG65A M- OJUSFLTL03T A
    1  /RONDADFLGG65A05XB203056510868  213011                NDADFLGG65A0000 000000
EDT?
```

4.45 ***Example 40:*** Use the **TAB** character to find all lines with the string **L03T A**, starting in column 40.

```
EDT?TAB 40
    9  /YBNDADFLGGE65NDADFLGG65A00JUSFLTL03T
EDT?0;# FIND "'L03T A"
    2  /YGDF54CA-ANI     NDADFLGG65A M- OJUSFLTL03T A
EDT?FIND
    4  /YGDF54CA-CR      NDADFLGG65A M- OJUSFLTL03T A
EDT?FIND
    6  /YGDF54DA-ANI     NDADFLGG65A M- OJUSFLTL03T A
EDT?FIND
    8  /YGDF54SP-COMB    NDADFLGG65A M- OJUSFLTL03T A
EDT?
```

Q. Generating a USE File for Use by SELEC Program

4.46 A USE file is a file of SELEC program commands. This file allows the user to execute the SELEC commands without inputting each and every command. This makes it easy to search for certain types of data base information which is required on weekly or monthly bases.

4.47 ***Example 41:*** Generate a USE file called **IOTSTX**.

```
? RUN:EDIT
EDT?ILINE
      ?SE:COFC=MIAMFLAEE44
      ?FI:MNGS
      ?SE:CLEAR
      ?SE:COFC=MIAMFLCAE55
      ?FI:MNGS
      ?SE:CLEAR
      ?SE:COFC=MIAMFLGRE35
      ?FI:MNGS
      ?SE:CLEAR
      ?SE:COFC=MIAMFLMEE32
      ?FI:MNGS
      ?.
.   1   SE:COFC=MIAMFLAEE44
EDT?0;# WRITE
.   1   SE:COFC=MIAMFLAEE44
.   2   FI:MNGS
.   3   SE:CLEAR
.   4   SE:COFC=MIAMFLCAE55
.   5   FI:MNGS
.   6   SE:CLEAR
.   7   SE:COFC=MIAMFLGRE35
.   8   FI:MNGS
.   9   SE:CLEAR
.  10   SE:COFC=MIAMFLMEE32
.  11   FI:MNGS
EDT?SAVE IOTSTX
.   1   SE:COFC=MIAMFLAEE44
EDT?END
?
```

5. SUMMARY OF TEXT EDITOR COMMAND FORMATS

5.01 A quick reference to the command formats for the SELEC program and text editor are presented as follows:

<n1;n2,n3> **APPEND** *<file name>*
COL *[device]*
*<n1;n2,n3>***COPY**
<n1;n2> **DELETE** *<string>*
DL *[device]*
[n] **EDIT**
END
[n1;n2] **FIND** *<character string>*
HELP *<mnemonic>,[device]*
[n] **ILINE**
LOAD *<device or file name>*
[n1;n2,n3] **MOVE**
PURGE *<file name>*
[n1;n2] **REPLACE**
[n1;n2] **SAVE** *<file name>,* **[UP]**
SET:LENGTH=*<n>*
SET:DC=*<character>*
SET:QUOTE=*<character>*
SET:*<label=character string>*
SHOW:LENGTH
SHOW:DC
SHOW:TAB
SHOW:$*n*
SHOW:LAST
SHOW:ALL
SHOW:QUOTE
TAB *[c] [n1,n2,n3...]*
[n1;n2] **WRITE** *[device [,***UN/LN***]]*
[n1;n2] **XCHANGE** *<search string>;<replacement string>*

6. REFERENCES

SECTION	DESCRIPTION
190-102-203	CAROT 2 Data Base Description and Input File Preparation
190-102-206	SELEC PROGRAM
190-102-207	CAROT 2 Report Generation Using the Report Program Generator (RPG) Program
865-203-100	CAROT 2 Engineering Considerations